

**In the Specification**

**Kindly replace paragraph [0032] with the following:**

[0032] The just described package creation process for Solaris does not include all of steps within the larger steps. A more complete understanding of the Solaris package building process can be found at the web site <http://docs.sun.com> or in the reference manuals for the relevant operating system. In the Sun Microsystems example, only the steps for creating a package using the Solaris native utilities ~~would be~~ were described. Each one of the other UNIX operating systems such as HP-UX and AIX, all have ~~there~~ their own native utilities for creating packages, and they ~~do~~ differ significantly, making it very difficult for a single person to be able to build packages that take advantage of the functionality offered by the various native utilities.

**Kindly replace paragraph [0033] with the following:**

[0033] The packager may be using the process of the present invention within a system 10 as shown in Fig. 1. The packager would be entering the necessary inputs (as will be explained) at computer or server 14 that contains the process 16. Server 14 may be part of a network 12 that includes servers or computers 18, 20, 22 and 24. The network could be any local area network (LAN) or wide area network (WAN) as is well known in the art. It is also contemplated that the network 12 could also be the internet and the servers shown could all be in different remote locations. The server 14 may also include an operating system database 26, which is used upon installation of the packages of the present invention to retain information concerning the package contents. This system 10 is shown, in this example, as having five servers, but this is simply for illustrative purposes, as the system 10 could have any number of servers. The servers 14, 18, 20, 22 and 24 can be running any one of the different UNIX operating systems. For example,

server 14 could be running the Digital UNIX operating system; server 18 could be running the Solaris operating system; server 20 could be running the HP-UX operating system; server 22 could be running the AIX operating system and server 22 could be running the Linux operating system. The process 16 may interact with the other servers 18, 20, 22 and 24 as will be described in order to provide the completed package. The packager<sub>1</sub> without any knowledge of the native utilities on each of the servers 14, 18, 20, 22 and 24<sub>1</sub> can build packages that utilize the full native functionality inherent in each of the different operating systems without knowledge of that native packaging utilities. To do this, requires processes or agents 16' to reside on each of the servers 14, 18, 20, 22 and 24, wherein communication occurs between processed 16 and 16'. For example when process 16 is invoked without communication to the other servers, a package native to the operating system of server 14 can be built.

**Kindly replace paragraph [0034] with the following:**

[0034] In another embodiment of the invention ~~may have a~~ master process 16 which, when appropriate will invoke or involve the other processes 16' on the other servers to obtain their native utility information. Processes 16 and 16' can communicate with each other across the network in a variety of ways well known in the art. In the present invention it is contemplated that sockets will be used to establish and maintain the connection between one or more processes. In this scenario, process 16<sub>1</sub> as the master process<sub>1</sub> would be responsible for accepting the inputs (as will be explained) from the user at server 14, and establishing a connection with another process in the network and issuing instructions to have that process build a package suitable for the operating system of the second process. This would be accomplished by the first process 16 recognizing that the operating system to be used is different than the one the process

Applicant: Derek Barrett  
U.S.S.N.: 09/747,737  
Filing Date: December 21, 2000  
EMC Docket No.: EMC-00-212

16 is running on. Upon this recognition, the process would contact and establish a connection with a process 16' that uses the particular operating system as input by the user. In this scenario the inputs from the user into process 16 would be sent via the connection to the process 16' which upon receipt of the needed inputs would build the package. It is contemplated that process 16' could either complete the building of the package, including local loading on the package onto installation media or send the package over the network back to process 16, which could load the package onto the installation media. It is also contemplated that each process in itself is a master process, which will allow a user to use the process to build packages from any one of the servers in the system.